

一.

(1) 编译器总体结构如下:

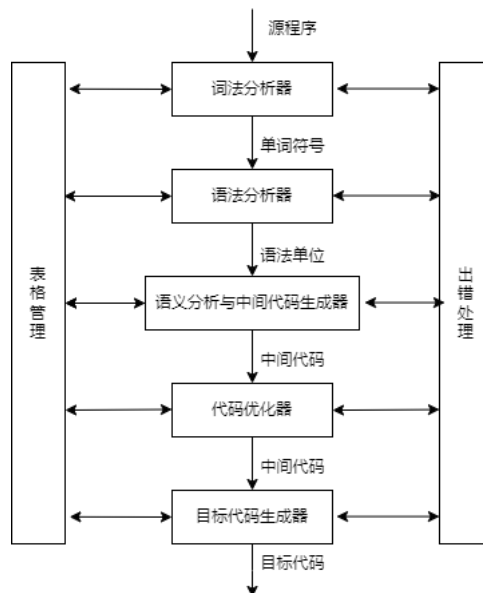


Figure 1: 编译器结构

- (i) 词法分析器将源代码的字符流转化为一系列词法单元 (token)，这些单元通常代表最小的语法元素，如关键词、运算符、标识符等，同时去除无用的空白符、注释，并报告词法错误。
- (ii) 语法分析器根据词法单元 (token) 的顺序检查源代码的语法结构，生成抽象语法树 (AST)。其目的是确保源代码符合语言的语法规则，并捕捉语法错误。
- (iii) 语义分析器会对 AST 进行语义检查，确保每个操作符、变量和函数调用的使用方式正确。中间代码生成器则将经过语义检查的代码转换为中间表示，这种表示比源代码更接近机器代码。
- (iv) 代码优化器通过分析和改进中间代码，去除冗余操作，提高执行效率。
- (v) 目标代码生成器将优化后的中间代码转换为特定硬件平台的机器代码或汇编代码。同时负责寄存器分配、指令选择和指令排布等任务。
- (vi) 表格管理模块负责管理编译过程中所需的符号表、类型表、常量表等信息，这些表格用于存储变量、函数和数据类型等信息。
- (vii) 出错处理模块负责在编译过程中处理各种错误 (如词法错误、语法错误、语义错误等)，并向用户报告有用的错误信息。

(2) T 型图如下:

最左推导:

$$\begin{aligned} E &\Rightarrow T \Rightarrow T * F \Rightarrow F * F \Rightarrow P * F \Rightarrow (E) * F \Rightarrow (E + T) * F \\ &\Rightarrow (T + T) * F \Rightarrow (F + T) * F \Rightarrow (P + T) * F \Rightarrow (c + T) * F \\ &\Rightarrow (c + F) * F \Rightarrow (c + P) * F \Rightarrow (c + id) * F \Rightarrow (c + id) * P \\ &\Rightarrow (c + id) * (E) \Rightarrow (c + id) * (E + T) \Rightarrow (c + id) * (T + T) \\ &\Rightarrow (c + id) * (F + T) \Rightarrow (c + id) * (P + T) \Rightarrow (c + id) * (id + T) \\ &\Rightarrow (c + id) * (id + F) \Rightarrow (c + id) * (id + P) \Rightarrow (c + id) * (id + c) \end{aligned}$$

最右推导:

$$\begin{aligned} E &\Rightarrow T \Rightarrow T * F \Rightarrow T * P \Rightarrow T * (E) \Rightarrow T * (E + T) \Rightarrow T * (E + F) \\ &\Rightarrow T * (E + P) \Rightarrow T * (E + c) \Rightarrow T * (T + c) \Rightarrow T * (F + c) \\ &\Rightarrow T * (P + c) \Rightarrow T * (id + c) \Rightarrow F * (id + c) \Rightarrow P * (id + c) \\ &\Rightarrow (E) * (id + c) \Rightarrow (E + T) * (id + c) \Rightarrow (E + F) * (id + c) \\ &\Rightarrow (E + P) * (id + c) \Rightarrow (E + id) * (id + c) \Rightarrow (T + id) * (id + c) \\ &\Rightarrow (F + id) * (id + c) \Rightarrow (P + id) * (id + c) \Rightarrow (c + id) * (id + c) \end{aligned}$$

任意推导:

$$\begin{aligned} E &\Rightarrow T \Rightarrow T * F \Rightarrow F * F \Rightarrow P * F \Rightarrow (E) * F \Rightarrow (E + T) * F \\ &\Rightarrow (T + T) * F \Rightarrow (F + T) * F \Rightarrow (P + T) * F \Rightarrow (P + F) * F \\ &\Rightarrow (c + F) * F \Rightarrow (c + P) * F \Rightarrow (c + id) * F \Rightarrow (c + id) * P \\ &\Rightarrow (c + id) * (E) \Rightarrow (c + id) * (E + T) \Rightarrow (c + id) * (T + T) \\ &\Rightarrow (c + id) * (F + T) \Rightarrow (c + id) * (P + T) \Rightarrow (c + id) * (id + T) \\ &\Rightarrow (c + id) * (id + F) \Rightarrow (c + id) * (id + P) \Rightarrow (c + id) * (id + c) \end{aligned}$$

(2) 对于 $id + id * (id - id)$,

最右归约:

$$\begin{aligned}(c + id) * (id + c) &\Leftarrow (c + id) * (id + P) \Leftarrow (c + id) * (id + F) \\&\Leftarrow (c + id) * (id + T) \Leftarrow (c + id) * (P + T) \\&\Leftarrow (c + id) * (F + T) \Leftarrow (c + id) * (T + T) \\&\Leftarrow (c + id) * (E + T) \Leftarrow (c + id) * (E) \\&\Leftarrow (c + id) * P \Leftarrow (c + id) * F \\&\Leftarrow (c + P) * F \Leftarrow (c + F) * F \\&\Leftarrow (c + T) * F \Leftarrow (P + T) * F \\&\Leftarrow (F + T) * F \Leftarrow (T + T) * F \\&\Leftarrow (E + T) * F \Leftarrow (E) * F \\&\Leftarrow P * F \Leftarrow F * F \\&\Leftarrow T * F \Leftarrow T \Leftarrow E\end{aligned}$$

最左规约:

$$\begin{aligned}id + id * (id - id) &\Leftarrow P + id * (id - id) \Leftarrow F + id * (id - id) \\&\Leftarrow T + id * (id - id) \Leftarrow E + id * (id - id) \\&\Leftarrow E + P * (id - id) \Leftarrow E + F * (id - id) \\&\Leftarrow E + T * (id - id) \Leftarrow E + T * (P - id) \\&\Leftarrow E + T * (F - id) \Leftarrow E + T * (E - id) \\&\Leftarrow E + T * (E - P) \Leftarrow E + T * (E - F) \\&\Leftarrow E + T * (E) \Leftarrow E + T * F \\&\Leftarrow E + T \Leftarrow E\end{aligned}$$

上述任意推导对应的规约:

$$\begin{aligned}
id + id * (id - id) &\Leftarrow id + id * (id - P) \Leftarrow id + id * (id - F) \\
&\Leftarrow id + id * (F - F) \Leftarrow id + id * (T - F) \\
&\Leftarrow id + id * (E - F) \Leftarrow id + id * (E - T) \\
&\Leftarrow id + id * (E) \Leftarrow id + id * P \\
&\Leftarrow id + id * F \Leftarrow id + F * F \\
&\Leftarrow id + T * F \Leftarrow id + T \\
&\Leftarrow P + T \Leftarrow F + T \\
&\Leftarrow T + T \Leftarrow E + T \\
&\Leftarrow E
\end{aligned}$$

对于 $(c + id) * (id + c)$,

最右规约:

$$\begin{aligned}
(c + id) * (id + c) &\Leftarrow (c + id) * (id + P) \Leftarrow (c + id) * (id + F) \\
&\Leftarrow (c + id) * (id + T) \Leftarrow (c + id) * (P + T) \\
&\Leftarrow (c + id) * (F + T) \Leftarrow (c + id) * (T + T) \\
&\Leftarrow (c + id) * (E + T) \Leftarrow (c + id) * (E) \\
&\Leftarrow (c + id) * P \Leftarrow (c + id) * F \\
&\Leftarrow (c + P) * F \Leftarrow (c + F) * F \\
&\Leftarrow (c + T) * F \Leftarrow (P + T) * F \\
&\Leftarrow (F + T) * F \Leftarrow (T + T) * F \\
&\Leftarrow (E + T) * F \Leftarrow (E) * F \\
&\Leftarrow P * F \Leftarrow F * F \\
&\Leftarrow T * F \Leftarrow T \Leftarrow E
\end{aligned}$$

最左规约:

$$\begin{aligned}(c + id) * (id + c) &\Leftarrow (P + id) * (id + c) \Leftarrow (F + id) * (id + c) \\&\Leftarrow (T + id) * (id + c) \Leftarrow (E + id) * (id + c) \\&\Leftarrow (E + P) * (id + c) \Leftarrow (E + F) * (id + c) \\&\Leftarrow (E + T) * (id + c) \Leftarrow (E) * (id + c) \\&\Leftarrow P * (id + c) \Leftarrow F * (id + c) \\&\Leftarrow T * (id + c) \Leftarrow T * (P + c) \\&\Leftarrow T * (F + c) \Leftarrow T * (T + c) \\&\Leftarrow T * (E + c) \Leftarrow T * (E + P) \\&\Leftarrow T * (E + F) \Leftarrow T * (E + T) \\&\Leftarrow T * (E) \Leftarrow T * P \\&\Leftarrow T * F \Leftarrow T \\&\Leftarrow E\end{aligned}$$

上述任意推导对应的规约:

$$\begin{aligned}(c + id) * (id + c) &\Leftarrow (c + id) * (id + P) \Leftarrow (c + id) * (id + F) \\&\Leftarrow (c + id) * (id + T) \Leftarrow (c + id) * (P + T) \\&\Leftarrow (c + id) * (F + T) \Leftarrow (c + id) * (T + T) \\&\Leftarrow (c + id) * (E + T) \Leftarrow (c + id) * (E) \\&\Leftarrow (c + id) * P \Leftarrow (c + id) * F \\&\Leftarrow (c + P) * F \Leftarrow (c + F) * F \\&\Leftarrow (P + F) * F \Leftarrow (P + T) * F \\&\Leftarrow (F + T) * F \Leftarrow (T + T) * F \\&\Leftarrow (E + T) * F \Leftarrow (E) * F \\&\Leftarrow P * F \Leftarrow F * F \\&\Leftarrow T * F \Leftarrow T \\&\Leftarrow E\end{aligned}$$

(3) $id + id * (id - id)$ 的最左语法树如下:

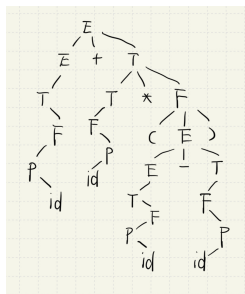


Figure 3: 语法树 1

$(c + id) * (id + c)$ 的最左语法树如下:

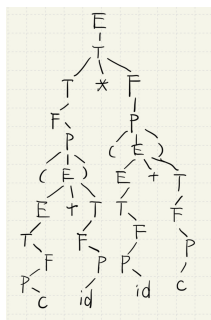


Figure 4: 语法树 2

(4) 对于 $id + id * (id - id)$,

短语为: $id, id, id, id, id - id, (id - id), id * (id - id), id + id * (id - id)$

简单短语为: id, id, id, id

句柄为: id

对于 $(c + id) * (id + c)$,

短语为: $c, c, id, id, c + id, id + c, (c + id), (id + c), (c + id) * (id + c)$

简单短语为: c, c, id, id

句柄为: c

三.

(1) 根据文法，列出方程组如下：

$$\left\{ \begin{array}{l} \langle \text{标号说明} \rangle = \text{LABEL} \langle \text{标号段} \rangle \\ \langle \text{标号表} \rangle = d \langle \text{标号段} \rangle \\ \langle \text{标号段} \rangle = d \langle \text{标号段} \rangle \\ \langle \text{标号段} \rangle = , \langle \text{标号} \rangle \\ \langle \text{标号段} \rangle = ; \\ \langle \text{标号} \rangle = d \langle \text{标号段} \rangle \end{array} \right.$$

求解得到正则表达式为

$$\text{LABEL } d(d|,d)^*$$

(2) 上述正则表达式对应的 DFA 如下：

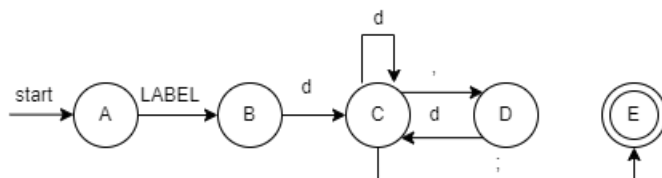


Figure 5: DFA

四.

(1)

$$S \rightarrow bES'$$

$$S' \rightarrow aES' | \epsilon$$

$$F \rightarrow S | a$$

$$E \rightarrow Fc$$

(2) First 集合如下:

$$\text{First}(S) = \{b\}$$

$$\text{First}(S') = \{a, \epsilon\}$$

$$\text{First}(F) = \{a, b\}$$

$$\text{First}(E) = \{a, b\}$$

$$\text{First}(a) = \{a\}$$

$$\text{First}(b) = \{b\}$$

$$\text{First}(c) = \{c\}$$

Follow 集合如下:

$$\text{Follow}(S) = \{\#, c\}$$

$$\text{Follow}(S') = \{\#, c\}$$

$$\text{Follow}(F) = \{c\}$$

$$\text{Follow}(E) = \{a, c, \#\}$$

(3) 由于

$$\text{First}(S) \cap \text{First}(a) = \emptyset$$

$$\text{Follow}(S') \cap \text{First}(aES') = \emptyset$$

故该文法是 $LL(1)$ 文法. 预测分析表如下:

	a	b	c	$\#$
S		$\rightarrow bES'$		
S'	$\rightarrow aES'$		ϵ	ϵ
E	$\rightarrow Fc$	$\rightarrow Fc$		
F	$\rightarrow a$	$\rightarrow S$		

五.

(1)

$$\langle \text{var_list} \rangle \rightarrow \text{id} \langle \text{var_list} \rangle'$$

$$\langle \text{var_list} \rangle' \rightarrow , \langle \text{var_list} \rangle' \mid \epsilon$$

(2) First 集合如下:

$$\text{First}(<\text{declaration}>) = \{\text{int}, \text{float}\}$$

$$\text{First}(<\text{type}>) = \{\text{int}, \text{float}\}$$

$$\text{First}(<\text{var_list}>) = \{\text{id}\}$$

$$\text{First}(<\text{var_list}>') = \{', ', \epsilon\}$$

$$\text{First}(\text{int}) = \{\text{int}\}$$

$$\text{First}(\text{float}) = \{\text{float}\}$$

$$\text{First}(\text{id}) = \{\text{id}\}$$

$$\text{First}(,) = \{,\}$$

Follow 集合如下:

$$\text{Follow}(<\text{declaration}>) = \{\#\}$$

$$\text{Follow}(<\text{type}>) = \{\text{id}\}$$

$$\text{Follow}(<\text{var_list}>) = \{\#\}$$

$$\text{Follow}(<\text{var_list}>') = \{\#\}$$

(3) 由于

$$\text{First}(\text{int}) \cap \text{First}(\text{float}) = \emptyset$$

$$\text{First}(, <\text{var_list}>) \cap \text{First}(\epsilon) = \emptyset$$

$$\text{First}(, <\text{var_list}>) \cap \text{Follow}(<\text{var_list}>') = \emptyset$$

故该文法是 $LL(1)$ 文法.

(4) 预测分析表如下:

	int	float	id	,	#
<declaration>	$\rightarrow <\text{type}><\text{var_list}>$	$\rightarrow <\text{type}><\text{var_list}>$			
<type>	$\rightarrow \text{int}$	$\rightarrow \text{float}$			
<var_list>			$\rightarrow \text{id}<\text{var_list}>'$		
<var_list>'				$\rightarrow ,<\text{var_list}>$	$\rightarrow \epsilon$

(5) 分析过程如下:

栈	输入缓冲	输出
#<declaration>	int x,y,z#	
#<var_list><type>	int x,y,z#	<declaration> \rightarrow <type><var_list>
#<var_list>int	int x,y,z#	<type> \rightarrow int
#<var_list>	x,y,z#	
#<var_list>'id	x,y,z#	<var_list> \rightarrow id<var_list>'
#<var_list>'	,y,z#	
#<var_list> ,	y,z#	<var_list>' \rightarrow ,<var_list>
#<var_list>	y,z#	
#<var_list>'id	y,z#	<var_list> \rightarrow id<var_list>'
#<var_list>'	,z#	
#<var_list> ,	,z#	<var_list>' \rightarrow ,<var_list>
#<var_list>	z#	
#<var_list>'id	z#	<var_list> \rightarrow id<var_list>'
#<var_list>'	#	
#	#	<var_list>' \rightarrow ϵ